**CODEIGNITER**

**1. Codeigniter URLs**

localhost/folder\_name/index.php/ctrlr\_name/fn // To run in browser

**2. To send data via URLs**

localhost/folder\_name/index.php/ctrlr\_name/fn?var1=val1&var2=val2

**3. Controller**

It's a class file & it associated with url

The class name must be same as file name & first letter of class name should be in uppercase

eg: class Example extends CI\_Controller

{

public function \_\_construct()

{

parent::\_\_construct();

}

public function index()

{

...... // code to execute

}

}

**4. Redirect fn/-**

To change the direction of one ctrlr to another ctrlr

redirect('url');

**5. Defining a Default Controller**

When url is empty the CI told to load a default ctrlr

in the path, application/config/routes.php need to write the below code

$route['default\_controller'] = 'ctrlr\_name';

**6. Private methods**

If we want to hide certain methods the function is declared in private or protected

this fn can't be served via url request

private function func\_name

{

.......

}

**7. Reserved Names**

CI alrdy using more keywords so it can't allow the user to same name

Like index, default, views etc

**8. Views**

It's a web page, like header, footer, sidebar etc

View can't directly loaded. It always loaded by ctrlr

$this->load->view('view\_name'); // this code written in ctrlr

$this->load->view('folder\_name/view\_name'); //if view file in the folder

$this->load->view('view\_name',$data); //if data need to send to view file

To load the view file the .php file extension does not need to be specified unless you use something other than .php.

**9. Models**

This used to access the db

In model page it written as,

Class Model\_name extends CI\_Model

{

Public function \_\_construct()

{

Parent:: \_\_construct();

}

Public function funct\_name()

{

………..

}

}

In controller it loaded as,

$this->load->model\_name(); // loaded in constructor fn

In ctrlr the model function is called as,

$this->model\_name->func\_name();

$this->model\_name->func\_name($data); // if model function needs any values from ctrlr

Model name rules are same as the ctrlr name rules

**10.** **example for send the data to view after getting value from model file in ctrlr**

class Blog\_controller extends CI\_Controller

{

public function blog()

{

$this**->**load**->**model('blog');

$data['query'] **=** $this**->**blog**->**get\_last\_ten\_entries();

$this**->**load**->**view('blog', $data);

}

}

**11. To store the data in db**

$this->model\_name->func\_name(data); // send the insert val to mdl frm ctrlr

$this->db->insert('table\_name', $data); // mdl file insert val in db

**12. Update the data in db**

$this->db->update('table\_name', $data);

**13.Select the data from db**

$this->db->select('\*');

$this->db->from(‘table\_name’);

$query = $this->db->get();

$result = $query->result();

return $result;

**14. Where condition**

$this->db->where(‘col\_name’,val);

**15. Helper**

The CI alredy consist some functions

In ctrlr the needed helper will call using below code

$this**->**load**->**helper('name'); // here name denotes helper name like url,session,date etc

There are 2 helper directory i. System helper ii. Application helper

When helper call first check in application if not exist then only check the system directory

To load multiple helpers

$this**->**load**->**helper( array('helper1', 'helper2', 'helper3') );

**16. Auto loading helpers**

If we need some helper throughout our application it’ll declared in autoload helper

To declare as auto load helper, This is done by opening the **application/config/autoload.php** file and adding the helper to the autoload array.

$autoload['helper'] = array('helper1’,’helper2’); // change in autoload.php file

**17. Libraries**

Path of the libraries in application folder is system/libraries/

To load library in ctrlr,

$this**->**load**->**library('class\_name'); // class\_name denotes pagination, form\_validation etc

To load more than one library,

$this**->**load**->**library(array('email', 'table'));

**18. Error Handling**

To generate errors,

show\_error*(*$message*,* $status\_code*,*$heading = 'An Error Was Encountered'*)*

**19. Web page caching**

When page is loaded the cache page is started and it store the frequently used datas or files and send to the requesting user’s browsers.

To enable cache,

$this**->**output**->**cache($n); // In any ctrlr method

To delete the cache values,

$this**->**output**->**delete\_cache();

## 20. Alternative Control Structures

## ‘ {‘ replaced by ‘: ‘ and ‘}’ replaced by endforeach;

If it is a if loop same as foreach but loop is closed by using endif;

<?php foreach ($todo as $item): ?>

<?=$item?>

<?php endforeach; ?>

**21. Calendaring Class**

To create dynamic calendar

To initialize the calendar class,

$this->load->library(‘calendar’); // In ctrlr constructor fn

To display the calendar,

Echo $this->calendar->generate();

Echo $this->calendar->generate(1994,5); // to display 1994 may

To passing data to calendar cells,

$data = array(

1 => ‘url1’, // if u click that date it go to the given url

2 => ‘url2’,

3 => ‘url3’ );

echo $this**->**calendar**->**generate(1994, 5, $data);

For setting display,

$prefs **=** array(

'start\_day' **=>** 'saturday',

'month\_type' **=>** 'long',

'day\_type' **=>** 'short'

);

$this**->**load**->**library('calendar', $prefs);

echo $this**->**calendar**->**generate();

**22. Email Class**

$this**->**load**->**library('email');

$this**->**email**->**from('your@example.com', 'Your Name');

$this**->**email**->**to('someone@example.com');

$this**->**email**->**cc('another@another-example.com');

$this**->**email**->**bcc('them@their-example.com');

$this**->**email**->**subject('Email Test');

$this**->**email**->**message('Testing the email class.');

$this**->**email**->**send();

To set the return path for undelivered mail,

$this**->**email**->**from('you@example.com', 'Your Name', 'returned\_emails@example.com');

To send multiple persons,

$this->email->to('one@example.com, two@example.com, three@example.com');

$this->email->to(array ('one@example.com', 'two@example.com'));

To attach file with mail,

$this**->**email**->**attach('/path/to/photo1.jpg');

**23. File Upload**

public function do\_upload()

{

$config['upload\_path'] = './uploads/';

$config['allowed\_types'] = 'gif|jpg|png';

$config['max\_size'] = 100;

$config['max\_width'] = 1024;

$config['max\_height'] = 768;

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('userfile'))

{

$error = array('error' => $this->upload->display\_errors());

$this->load->view('upload\_form', $error);

}

else

{

$data = array('upload\_data' => $this->upload->data());

$this->load->view('upload\_success', $data);

}

}

**24. encode, decode :**

base64\_encode();

base64\_decode();

**25. date picker:**

$(document).ready(function () {

$("#start\_date").datepicker({

dateFormat: 'yy-mm-dd',

minDate: 0 // to hide past date

});

});

**26. split and join values**

explode(",",$val) ; //to split

implode(separator,array); // to join

**27. To autocomplete**

<input name="location" type="text" autocomplete="off" list="tourloc" id="location" />

<datalist id="tourloc" >

<?php

foreach($name\_lists as $name\_list)

{

echo "<option>".$name\_list->location."</option>" ;

} ?>

</datalist>

**28. To get the value from html to script:**

var value= document.getElementById(id\_mentioned\_in\_html\_tag).value;

**29. To ajax call:**

$.ajax({

url: 'http://localhost/codeigniter/index.php/tour/searchdb', // path to ctrlr

data: { id1:id1,id2:id2}, // eg: data: {name:name, pwd:pwd};

type: 'POST',

success: function(data) {

$("#id").html(data); // result got from that given result iff success

}, // id- where the o/p need to display & in html id is mentioned

Eg: <div id=”id”>

error: function() {

alert('Your searching result is not found');

}

});

**30.To go another page using script:**

window.location="needed link";

**31. To upload multiple files:**

Getting files from HTML,

<input type="file" name="userfile[]" id="file" multiple=""/>

To store in folder,

$files = $\_FILES;

$cpt = count($\_FILES['userfile']['name']);

for($i=0; $i<$cpt; $i++)

{

$\_FILES['userfile']['name']= $files['userfile']['name'][$i];

$\_FILES['userfile']['type']= $files['userfile']['type'][$i];

$\_FILES['userfile']['tmp\_name']= $files['userfile']['tmp\_name'][$i];

$\_FILES['userfile']['error']= $files['userfile']['error'][$i];

$\_FILES['userfile']['size']= $files['userfile']['size'][$i];

$config = array();

$config['upload\_path'] = 'D:\xampp\htdocs\images\tour\upload'; //give the path to upload

In folder

$config['allowed\_types'] = 'gif|jpg|png';

$this->load->library('upload', $config);

$fileName = $\_FILES['userfile']['name'];

$images[] = $fileName;

$data['fileName'] = implode(',',$images);

$this->upload->do\_upload();

}

**32. To set the session values:**

$this->session->set\_userdata('user\_id',$user\_id);

**To clear session values assigned by the user:**

$this->session->unset\_userdata('user'); // here everything constant

**To clear particular values only,**

$this->session->unset\_userdata(‘that\_particular\_var\_name’);

**To Retriving all session datas:**

$this->session->all\_userdata(); // it returns as associative array

**To Retriving particular session data:**

$user\_id = $this->session->userdata('**user\_id**');

**33. To set the rules for form validation:**

$this->form\_validation->set\_rules('name', 'name', 'required');

if($this->form\_validation->run() == true)

{

// code to execute if form satisfy the al validation rules

}

else

{

$this->load->view('form'); // if form not satisfy again go to same form with error

}

In HTML side,

<input type="text" name="name" value="<?php echo set\_value('name'); ?>" />

<span class="error">\* <?php echo form\_error('name'); ?></span>

**34. To send the data with redirect function:**

$this->session->set\_flashdata(Variable,$value);

redirect();

**To get the value from redirect function:**

$variable = $this->session->flashdata(Variable);

**35. Join Query:**

$this->db->select(“\*”);

$this->db->from(‘blogs’);

$this->db->join(‘comments’,’comments.id=blogs.id’);

// join the id in comments table with blogs table using id column in blogs

**36.**